REMARKS

In the outstanding office action referenced above, the examiner has made some objections to the drawings. The attached proposed drawing changes address these objections. In particular, Applicant proposes to change Figures 1(a) and 1(b) to add the legend "prior art." With regard to Figure 2(b), Applicant proposes to change the sub "0" that appears in the step 14 box to sub "N." The examiner will also note that Applicant has amended line 19 of page 5 to change the incorrect sub "M" to the correct sub "N." Finally, Applicant has added boxes to the arrows in Figure 3 to indicate that an arrow pointing to a box indicates data is inputted to the process represented by the box, whereas data output from a process is indicated by an arrow pointing away from the box.

Applicant has also redone some of the lines, numbers and letters to make them totally legible. Applicant submits that these various drawing changes overcome all of the outstanding objections and respectfully requests the examiner to enter them and withdraw the objections.

There are numerous objections to the specification. Applicant has amended the specification in numerous places to address all of these objections and will not describe them in further detail. Applicant should mention, however, that the abstract has been rewritten to meet the requirements set forth by the examiner in paragraph 6 of the outstanding office action.

With regard specifically to the objection raised in paragraph 26 of the office action, Applicant informs the examiner that the probability factor should be X^{-N}. Page 6 of the text has been amended to show this. This means that the description and claims 3 and 4 are now consistent, and this should overcome the enablement rejection under 35 U.S.C. §112, first

paragraph, set forth in paragraph 42 of the office action. Since the correct probability factor was set forth in claims 3 and 4 as filed, no new matter has been introduced.

The probability factor is the selection probability when forming a randomly selected fault list from the original fault list, as described on page 3 of the specification, at lines 7 and 8. Each individual fault has this chance of being selected for the subset. The value of N increases with each iteration of the process as described on page 8, at line 14. The value of X is typically 2 but can either be smaller or larger. A smaller value of X will make the method slower but will give a more accurate result. A larger value will make the method faster but give a less accurate result.

The effect of changing the probability on the quality of the final ordering can be thought of in terms of a mechanical analogy. The analogy relates to passing a bag of rocks of mixed sizes through a number of sieves to put them into piles of different sizes. The process starts with a sieve with a very large mesh to pass all except the largest rocks that are in the first pile. The mesh size is steadily reduced to form the successive piles. If only a few different mesh sizes of widely differing sizes are used then the process is faster, but each pile will contain a larger range of sizes. If more sleeves are used with a closer range of mesh sizes then the process will take longer but each pile of rocks will contain a smaller range of sizes.

As regards the examiner's comments in paragraph 36, methods for "forming a vector set" are very well known to a person skilled in the art, and so this phrase has a well-understood technical meaning. The publication credited with being the original description of such a technique is referenced on page 2 of the application as filed. The inventor has commented that it is widely described in many other papers and textbooks on this subject.

Hence, Applicant submits that there is no need to provide a further technical description of this phrase.

The examiner has rejected claim 1 under 35 U.S.C. §103(a) as obvious over U.S. Patent No. 5,983,381 to Chakradhar in view of U.S. Patent No. 6,052,809 and Rohrbaugh (U.S. Patent No. 6,067,651).

The Chakradhar patent describes a reordering method based on the simulation of subsets of faults. The fundamental difference between the approach taken by Chakradhar and the invention claimed in the present application is that Chakradhar uses subsets of vectors that are selected on a deterministic basis by actual fault simulation, rather than random simulation. In contrast, the subsets of the present invention are randomly selected. Accordingly, Chakradhar does not teach this critical aspect of Applicant's claimed invention.

This omission is disclosed in neither of the two secondary references. With regard to Bowden, the reference describes a method for generating vectors. It does not present a method of reordering the vectors into a near optimal ordering. The reordering is fundamental to the invention described in the present application. This is not disclosed in Bowden, or in any of the other prior art of record.

With regard to Rohrbaugh, the reference describes a method of dynamic compaction, where the compaction takes place during the generation process. It reduces the volume of vectors generated but does not guarantee that the final set of vectors is optimally ordered. One of the fundamental differences between Rohrbaugh and the method claimed in the present application is that the claimed invention is a method for ordering vectors after the generation process on the complete vector set. In contrast, Rohrbaugh describes a method for compacting during the generation process.

A further difference between Rohrbaugh and the present invention is that Rohrbaugh does not include any teaching for placing the vectors in the set in an optimal order. Instead, this document teaches a method only for reducing the number of vectors. Furthermore, there is no teaching in Rohrbaugh of a method that uses randomly selected subsets of faults. This is a key step in the method in which the present invention is embodied.

In view of the above, combining these three references in the manner suggested by the examiner would not teach the subject matter of claim 1. Furthermore, combining Chakradhar and Bowden would not disclose, teach or suggest the subject matter of claim 6 for the same reason, i.e., the combination fails to disclose numerous features of the invention as claimed in claim 6.

Accordingly, Applicant submits that claims 1 and 6 are patentable over the prior art of record and respectfully requests the examiner to withdraw both obviousness rejections.

Applicant notes that the office action does not contain specific prior art rejections of claims 2-5 and 7. However, the examiner also does not indicate they are allowable. Applicant submits that, because the allowability of claim 1 has been shown, and since all of the other claims depend ultimately upon claim 1, claims 1-7 are patentable over the prior art of record and Applicant respectfully requests expeditious notice thereof.

In short, Applicant submits that all of the outstanding objections and rejections have either been overcome or shown to be inappropriate. Should the examiner have any questions

or comments concerning the above-referenced application, he is respectfully requested to contact the undersigned attorney at the office number below.

Respectfully submitted,

Judith M. Riley
Registration No. 31,561
Gifford, Krass, Groh, Sprinkle,
Anderson & Citkowski, P.C.
280 N. Old Woodward Ave., Suite 400
Birmingham, MI 48009-5394
(248) 647-6000

Attorney for Applicant

JMR/gs
Enclosures

GS–W:\Word Processing\jmr\caf27202-amd.doc

CERTIFICATE OF MAILING BY "EXPRESS MAIL"

"EXPRESS MAIL" MAILING LABEL NUMBER EL 983 170 185 US

DATE OF DEPOSIT October 7, 2003

I hereby certify that this paper or fee (along with any paper referred to as being attached or enclosed) is being deposited with the United States Postal Service "Express Mail Post Office To Addressee" Service under 37 CFR 1.10 on the date indicated above and is addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Lynn Hill

ABSTRACT

~~IMPROVED METHOD FOR TESTING FOR THE PRESENCE OF FAULTS IN DIGITAL CIRCUITS~~

A method of testing for the presence of faults in digital logic circuits is described. The method involves re-ordering a number of test vectors for testing digital circuits by selecting faults at random from an original fault list to form a sample fault list F_N and then forming a vector set T_{N-1} and then simulating the vector set T_{N-1} against the fault list F_N. Any vector from the set T_{N-1} which does not detect any fault is discarded and the remaining vectors are saved as vector set T_N. The method steps are repeated N times (with N having a value of 1 to M. Duplicated vector patterns in each vector set are removed and then the final vector set is initialised to produce a final vector set T_F.

~~Embodiments of the invention are described.~~

IMPROVED METHOD FOR TESTING FOR THE PRESENCE OF FAULTS IN DIGITAL CIRCUITS

The present invention relates to a method of testing for the presence of faults in digital logic circuits. In particular, the invention also relates to a method for re-ordering "test vectors" to achieve an improved ordering which maximises the "fault coverage" on a digital circuit which is reached with a limited number of test vectors.

The testing of faults in digital circuits using test vectors is well known. This is typically done by applying a test vector, which is a set of binary values, to a digital circuit on either the primary inputs of the ~~circuits~~ circuit or via a "scan chain" connecting the state elements, which are normally flip-flops, within the digital circuit and expected response values on the circuit primary inputs are captured within the state elements, prior to being accessed via the scan chain. The term [["]]fault coverage[["]] is defined as a measure of the number of faults within the circuit which an individual test vector or set of test vectors will detect. The common fault model is "stuck-at" faults which are represented by short-circuits to power or ground on the primary inputs or outputs of logic gates within the circuit. Fault coverage is measured using a fault simulator which determines whether the output response of the circuit is affected.

A large number of existing methods for generating test vectors [[is]] are known and are published in the public domain. The best known is the Roth D Algorithm, originally published in 1966 and J.P. Roth "Diagnosis of Automota Failures: A Calculus and a Method", IBM Journal of Research and Development, Vol 10, No.4,pp.278-291, July, 1966.

Fig. 1 depicts an example of a test vector comprising two scan chains, which test vector is a list of binary values to be applied to the digital circuit under test via the primary input or via a scan chain. A scan chain is a method of serially loading values into the flip-flops in the digital circuit. This is a well known and widely published design technique. In the prior art example shown, two scan chains are depicted, although any number of scan chains may be used.

Existing methods of testing for faults on an integrated digital circuit suffer from the principal disadvantage that a very large amount of test vectors is required in order to provide a satisfactory fault coverage. This can take a relatively long time and requires a relatively large amount of memory and is a limiting factor in the design and testing of such circuits.

An object of the present invention is to obviate or mitigate at least one of the above-mentioned disadvantages.

This is achieved in the broadest sense by using randomly selected fault lists to select subsets of test vectors.

According to one aspect of the present invention, there is provided a method of maximising the fault coverage on an integrated digital circuit by re-ordering a number of test vectors for testing the digital circuit, said method comprising :

a) providing an initial set of test vectors T_0;

b) providing an original set of faults F_0;

c) selecting faults at random from the original fault list to form a sample fault list F_N;

d) forming a vector set T_{N-1} and simulating the vector set T_{N-1} against fault list F_N;

e) discarding any vector from the vector set T_{N-1} which does not detect any faults~~, and~~ ;

f) saving the remaining vectors as vector set T_N[[,]];

g) repeating the above steps ~~a) to e)~~ c) to f) N times with N having a value of 1 to M so that at the end of ~~N~~ M steps, ~~saving~~ test vectors ~~T_0~~ T_1 to T_M are saved;

[[g]] h) removing duplicate vector patterns in each vector set T_N; and

[[h]] i) initialising the final vector set and appending vector sets V_M ~~to~~ through V_0 to produce a final vector set T_F.

Preferably, in ~~steps a) to f)~~ step g) M is 10 and ~~these~~ steps c) to f) are therefore repeated ten times.

Preferably, the method of removing duplicate vector patterns is achieved by :

[[i]] j) copying the original fault list F_O to provide a secondary fault list ~~G_0~~ G_N;

[[j]] k) fault simulating vector set T_N against G_N and deleting any vectors which find no faults;

[[k]] l) saving the resulting vectors as vector set V_N and saving the list of undetected faults as list G_{N-1};

[[l]] m) repeating ~~step g) to i)~~ steps k) and l) M+1 times with N having values M to 0;

These and other aspects of the invention will become apparent from the following description, when taken in combination with the accompanying drawings, in which :

~~Fig. 1 is a depiction of a test vector;~~

Fig. 2 is a flow chart of the steps involved in the method of re-ordering test vectors for maximising the fault coverage on a digital circuit;

Fig. 3 is a legend to terms used in the flow chart of Fig. [[1]] 2, and

Fig. 4 is an example of a graph of fault coverage showing the number of faults detected against the number of vectors for the original vectors and vectors after using the new method/algorithm.

Reference is first made to Figs. 2 and 3 of the drawings which depict a flow chart of a sequence of steps involved in re-ordering test vectors to provide a test vector generation pattern for maximising the fault coverage on an integrated digital circuit using a limited number of test vectors.

In stage 1, (steps a) to f)) an initial set of vectors are provided (step a) and these are copied to form a set of test vectors T_O to place these vectors in a near-optimal order for detecting a set of faults F_O. Stage 1 has two principal steps. In the first step (step 10), a list of faults is selected at random with a probability of 2^{-N} from the original fault list F_O to form a sample or subset fault list F_N. The second major step (step 12) in stage 1 is fault simulating the vector set T_{N-1} repeatedly against the fault list F_N and then discarding any vector which does not detect any faults. In step 12, the ordering of individual vectors within the vector set are

alternately reversed and randomised. The resulting vector set, that is those that do result in the finding of faults, is saved as vector set T_N.

The two major steps 10 and 12 are repeated [[N]] M times ~~with N taking the value 1 to M~~ where M is 10 in this example.

At the completion of the tenth step 12, there are ten sets of vectors saved, T_1 to T_{10}.

At the end of the stage 1, the method involves stage 2 wherein duplicate vector patterns are removed from the vector pattern list T_N. In this case the original fault list F_O is again copied and denoted as fault list ~~G_M~~ G_N (step 14). In the next step in stage 2, step 16, the vector set T_N is fault simulated against the fault list G_N and any vectors which result in no faults being found are deleted. After the fault simulation, the resulting vector set is saved as V_N and the list of undetected faults is saved as G_{N+1}. Stage 2 is repeated M plus 1 times, with ~~M~~ N taking values M down to zero where M is 10.

The final stage in the methodology is stage 3 in which the final vector set is initialised (step 18) and vector sets V_M to V_O are appended together to produce a final vector set T_F (step 20).

The re-ordering of the generated test vectors in the way described above allows a digital integrated circuit to be tested much quicker than with the prior art test vector ordering, such that a digital circuit can be tested in typically one tenth of the time using the prior art re-ordering test vector set. This means that much less memory is required and the design process is speeded up, resulting in a considerable economic benefit.

Fig. 4 depicts a graph of faults detected against a number of vectors. It will be seen that a larger number of faults are detected for vectors re-ordered after the new re-ordering method compared to an original number of vectors where there is less than about 700 vectors used. This increase is most dramatic for a lower number of vectors, such that this re-ordering algorithm maximises the fault coverage for a lower number of vectors when testing an integrated digital circuit.

Various modifications may be made to the re-ordering methodology hereinbefore described, without departing from the scope of the invention. For example, the repetition of each stage may take values in excess of or less than 10, although this will have an effect on the time taken to test the digital circuit. In addition, the probability factor of ~~$(X^{-N})\ 2^{-M}$~~ X^{-N} in this example may be varied. Typically, the value of X is 2. Increasing the value of X from 2 will reduce the time taken for the re-ordering but will reduce the quality of result and decreasing the value X from 2 will result in a longer time for re-ordering but will be more accurate. In addition, the duplicate vector patterns in the vector sets are removable by an alternative method wherein a text search is conducted through the list of files of vector patterns to look for identical vector patterns and once the identical vector patterns have been identified they are deleted. It will be appreciated that after re-ordering the vectors, a larger number of faults are detected by any size of subset of the original vectors than by the original vectors themselves and, as indicated above, this significantly reduces the test time for the digital circuit and requires less memory capacity in the test apparatus, resulting in a more effective and more efficient test system.

```
SCAN_TEST =

    pattern = 0;
    apply "grp1_load" 0 =
        chain "chain_0" =
"11111101011100001111100011100110110101000001010010\
10001100000111000110100001101001110010001010011110\
11011100011000010001100011110010001001010100000011\
10101010010011001011011001011001100100000110010010\
01110100000000111101111011001100111010111011010010\
00100111111001010110100001001100000011000011101101\
10101111011011010001101100111001000110100011011100\
10101111100100000001110110010000110011000011111000\
00010010001010110011101110001111111111100111100111\
01011000111001111101000110001111011101110011100111\
00100011010101100101010111000101010101111010011000\
00110111110111011101111001110100010010001000011111\
11101100010011101101111001110111111011011100110011\
11110111000110000101111100001000011100111111101111\
01010000100000010001101010101111101001110110110100\
01111011111011111111101001101011001110001100010100\
11011110111100100001000010111111111110000010110011\
10010011001110011111010100000011010010100110001101O\
11100000101011010110101001011011111111010011110001\
00011010011111110110001010011011010101101010010O110\
10111110101011100010000100111000011000011000010111\
01101101100000110011010011000011011000001011110110\
11010011011110000110001001110001110110101001000110\
11000001101111111111011110111010010001101010000011\
10110110010001011000011000100111101010111111011001\
11110001010001000110001110011101011100001000100000\
10110110111010010011001001110110010010100010110111\
10011100101100110110010011000001010110101011110100\
01010000101101101100100101011000010011011100010110\
00111110001010111110000000100101000011011000001100\
                    000111011100100001110010000010110001110000l";
```

Prior Art

Fig. 1(a)

```
chain "chain_1" =
"10111100101010100011111100111000110110011011001111\
01110011101001100111111011000011101010101011111110110\
00111010001000110110101011010101010010010000100101\
01001010100001101001100100110110000010011000001010\
01010011100000000101010110110000111111010001010001\
01011000101011110011101100100110011111010001101100\
00111111100000011101100000001110011111011001100110\
11011100000000001000101011100001111110001101000110\
11011001000000011000111010001011101110011010101111\
00010001010110110010011111111011001000001111111000\
11100100011101110011011001011011000110111011001100\
00110001100010110111110100101110101110000101101001\
00011011010100111101001001011010100111011000110101\
01010000001100110100000011111100100101000001100011\
00000111010111011000010100011100011001110100010101\
01000010000000000110111100010010011001010000000111\
01010001110101000001000100000011000011000001010101\
10100011011011110000001100000110110010011101111110\
01001000101101000101110110111101000000111010000010\
10001101111100100001110101110110101101000011101110\
10110000101000110100111101100100100011110100011111\
00101011111000000111101110111101010001101000100011\
00010011001110110100001101110000101001110111110110\
11000001100101110110101101010011110101101000001111\
01001111011000111111100001001110001100110001011000\
10101110101101010100000010100010111111101110111011\
10001110010010100100001101001101100011001011110101\
01000010000000111101000111101100010010000000101001\
11111110100110001011011110101000001010001110000001\
11001010001101100100101101101100011011111110011010\
01111100011111110000010011000110111011000000";
```

Prior Art

Fig. 1(b)

STAGE 2

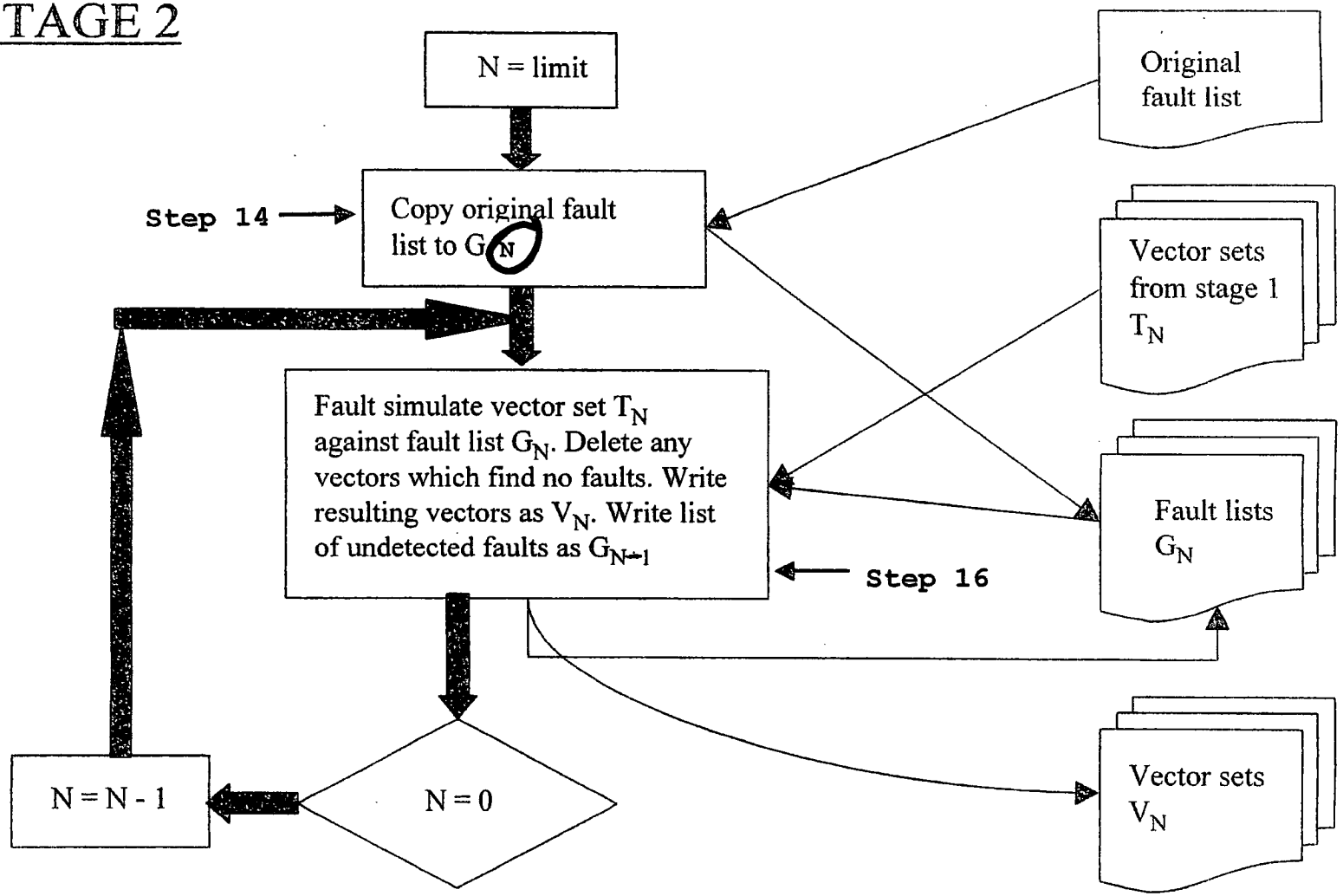
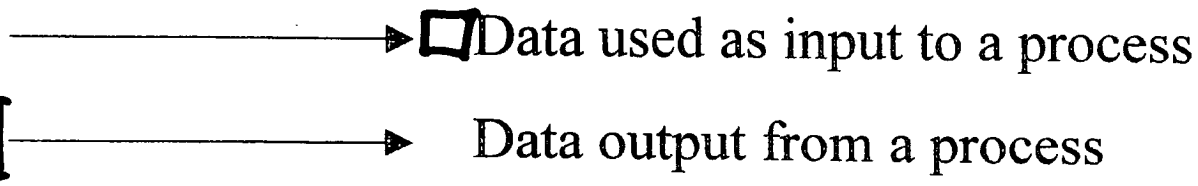


Fig. 2(b)

Key to flow diagrams



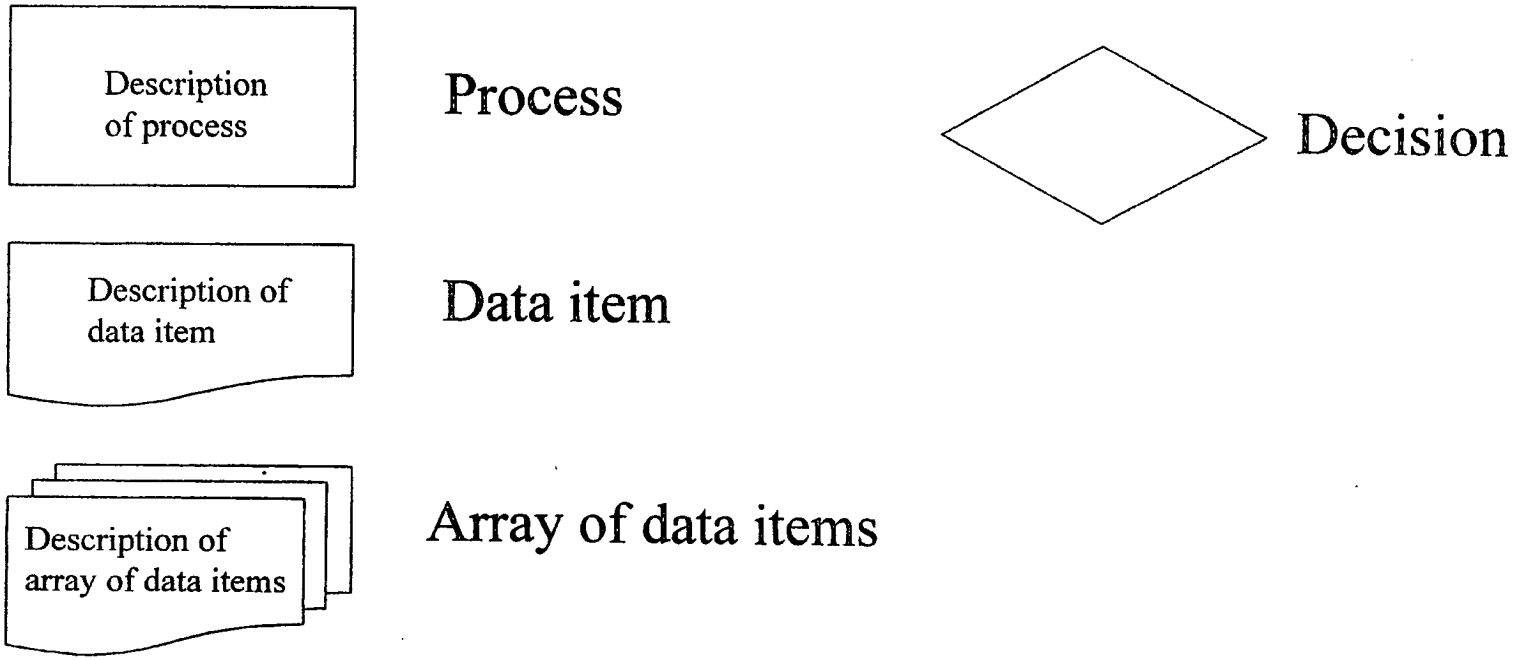


Fig. 3